

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

3D UŽIVATELSKÁ ROZHRANÍ

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

TOMÁŠ HRADEČNÝ

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ
FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

3D UŽIVATELSKÁ ROZHRANÍ 3D USER INTERFACES

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

TOMÁŠ HRADEČNÝ

VEDOUCÍ PRÁCE
SUPERVISOR

ING. PAVEL ŽÁK

BRNO 2011

Abstrakt

Tato bakalářská práce se zabývá 3D uživatelskými rozhraními. Jsou v ní představeny některé existující přístupy k těmto rozhraním. Dále uvádí různé druhy vstupních a výstupních zařízení sloužících pro interakci člověka s počítačem. Hlavním předmětem práce je návrh, implementace a testování aplikace demonstrující koncept prohlížeče hierarchických struktur, který je předveden na souborovém systému. K jejímu vytvoření byl využit soubor nástrojů OpenSceneGraph. Nakonec jsou zde také zmíněny ohlasy uživatelů na toto rozhraní a možnosti jeho dalšího vývoje.

Abstract

This bachelor's thesis deals with the area of 3D user interfaces. It introduces several existing approaches to this topic as well as various types of I/O interaction devices. The main goal of this work was to design and implement concept 3D interface for hierarchical structures browser, that was demonstrated as a file-browser application. Final application was developed using OSG, undertaking simple user experience evaluation.

Klíčová slova

3D uživatelské rozhraní, OpenSceneGraph, prohlížeč hierarchických struktur, graf scény, uživatelská interakce.

Keywords

3D user interface, OpenSceneGraph, hierarchical structures browser, scene graph, user interaction

Citace

Hradečný Tomáš: 3D uživatelská rozhraní, bakalářská práce, Brno, FIT VUT v Brně, 2011

3D uživatelská rozhraní

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Pavla Žáka. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Tomáš Hradečný
12.5.2011

Poděkování

Chtěl bych poděkovat svému vedoucímu Ing. Pavlu Žákovi za jeho ochotu a pomoc při vytváření této bakalářské práce.

© Tomáš Hradečný, 2011

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

Obsah.....	1
1 Úvod.....	3
2 Uživatelská rozhraní	4
2.1 Grafická uživatelská rozhraní	4
2.2 3D uživatelská rozhraní.....	5
2.2.1 Historie	5
2.2.2 Možnosti ovládání.....	6
2.2.3 Výstupní zařízení	8
2.3 Existující přístupy k 3D UI.....	10
2.3.1 Cooliris	10
2.3.2 BumpTop 3D Desktop.....	11
2.3.3 Prohlížeče souborových systémů	12
2.3.4 Blízká „budoucnost“	13
3 OpenSceneGraph	14
3.1 Graf scény	15
3.2 Komponenty	15
4 Návrh rozhraní.....	17
4.1 Volba vhodného rozhraní	17
4.2 Stromová struktura prohlížeče.....	18
4.3 Prohlížeč formou zdi souborů.....	19
4.3.1 Ovládání kamery ve scéně	19
4.3.2 Detaily souborů	20
4.3.3 Jiné možnosti zobrazení.....	20
5 Implementace aplikace.....	21
5.1 Sestrojení grafu scény	21
5.2 Transformace objektů.....	21
5.3 Běh aplikace	22
5.3.1 Průchody grafem scény	22
5.3.2 Animace.....	22
5.3.3 Obsluha událostí.....	23
5.3.4 Dynamické vytváření a rušení objektů	23
5.3.5 Hierarchie souborů	24
6 Testování	25
6.1 Uživatelské testy	25

6.2	Zhodnocení	26
7	Závěr	27
	Literatura	28
	Seznam příloh	29

1 Úvod

Tato bakalářská práce pojednává o 3D uživatelských rozhraních, jejich možnostech a využitích. Trojdimenzionální rozhraní obecně jsou stále více v oblibě, nicméně zatím jejich počet není příliš vysoký. Motivací pro tuto práci proto bylo vytvoření netradičního a pokud možno efektivního řešení, které by se pokusilo oprostit od zažitých přístupů, které zde již po mnoho let přetrvávají téměř nezměněny od doby, kdy se poprvé objevil koncept grafického uživatelského rozhraní, tak jak jej známe dnes, tedy klasické rozmístění prvků, manipulace s okny, ikony a podobně. Dalším důvodem pro tvorbu 3D rozhraní jsou stále se zdokonalující technologie nejen vstupních zařízení, které umožňují uživatelskou interakci mnoha způsoby, jako kupříkladu sledováním uživatele nebo jeho mozkové činnosti, ale také výstupních v podobě 3D monitorů, projektorů a různých druhů displejů. Byl tedy kladen důraz na to, aby výsledné rozhraní umožňovalo uplatnění těchto prostředků. Hlavním cílem práce byl návrh a implementace aplikace, která by demonstrovala možnost zobrazování hierarchických stromových struktur, konkrétně prohlížeč souborového systému.

Druhá kapitola této práce popisuje základní aspekty uživatelských rozhraní, nejprve obecná grafická a poté 3D rozhraní, přičemž je zmíněna historie, možnosti jejich ovládání pomocí širokého spektra metod a také technologie výstupních zařízení. Dále jsou představeny některé zajímavé existující přístupy k rozhraním využívajících třetí rozměr pro zobrazování galerií obrázků nebo pro prohlížeč souborových systémů. Další kapitola se věnuje grafickému souboru nástrojů *OpenSceneGraph*, ve kterém byla výsledná aplikace implementována. Jsou zde popsány jeho základní rysy, komponenty a také princip grafu scény, který je pro tento nástroj stěžejním prvkem. Čtvrtá kapitola pojednává o celkovém návrhu ukázkové aplikace počínaje volbou vhodného rozhraní. Rozebrány jsou dva konkrétní návrhy, jeden v podobě stromu souborů a druhý formou zdi, který byl nakonec implementován. Jsou nastíněny některé aspekty tohoto rozhraní a jeho možnosti. V páté části se potom nachází bližší popis algoritmů použitých při vývoji. Mimo jiné způsob, jakým je tvořen graf scény a jakým se manipuluje s objekty. Dále se řeší samotný průchod grafem a s ním spojené animace jednotlivých elementů rozhraní, obsluha událostí vyvolaných interakcí uživatele, problematika dynamického vytváření a rušení objektů ve scéně a také postup pro získání reálné hierarchické souborové struktury. Poslední kapitola se zabývá testováním výsledného rozhraní přímo uživateli a shrnuje jejich reakce a celkové zhodnocení.

2 Uživatelská rozhraní

Rozhraní zajišťuje přenos dat mezi zařízeními nebo programy a určuje způsoby, jakými tyto entity mezi sebou komunikují. Hardwarovým rozhraním je například sběrnice, softwarovým je mimo jiné právě grafické uživatelské rozhraní.

2.1 Grafická uživatelská rozhraní

Grafická uživatelská rozhraní (dále jen GUI – *graphical user interfaces*) podporují interakci s uživatelem ne jen formou textového zobrazení, ale také grafickými prvky, které ji činí přehlednější a efektivnější. V dnešní době je každý zvyklý na ovládání počítače pomocí kurzoru myši, kliknutím na ikony a manipulací s okny. Ne vždy tomu však takto bylo, ve skutečnosti byla cesta vývoje velmi dlouhá a komplikovaná. Tato část vychází ze zdroje [5].

Za otce GUI je považován Douglas Englebart, který v roce 1962 publikoval práci Rozšíření lidského intelektu (v originále *Augmenting human intellect*¹), ve které zmiňuje své názory na zefektivnění uživatelské interakce. Tvrdil, že počítače nemají nahradit lidský intelekt, ale spíše jej podpořit a obohatit. Tuto teorii vysvětloval na příkladu architekta navrhujícího budovu v podobné formě, jakou dnes nabízí moderní CAD software. Svými názory ovšem poněkud předběhl dobu, jelikož tehdy existovaly pouze sálové počítače, které byly svým výkonem absolutně nedostačující. O několik let později Englebart se svým týmem demonstroval svou práci na systému nazvaném *oN-Line System*, který se skládal z malého displeje zobrazujícího vektorovou grafiku a třech vstupních zařízení. Těmi byla jednak klasická klávesnice, dále pak klávesnice s pěti tlačítky, která tedy poskytovala 32 různých možností vstupu, a také jedna z prvních myší. Spolu s vynálezem myši se objevil i kurzor v podobě šipky, který známe dodnes. Lidé v té době většinou nedokázali pochopit ani si představit věci jako e-mail, sdílení souborů po síti, hypertextové odkazy a další věci, které se pokoušel na svém systému přiblížit veřejnosti.

Těchto konceptů si všimlo také vedení firmy Xerox a rozhodlo se podporovat rozvoj této technologie. Jedním z jejich prvních vynálezů byla laserová tiskárna. K tomu, aby ovšem mohly být dokumenty pro tisk upraveny, bylo vhodné kvalitní grafické rozhraní. Jelikož v této době neexistoval počítač vhodný pro tento účel, Xerox se rozhodl vyvinout svůj vlastní. Jeho rozměry byly natolik malé, že se vešel pod stůl. Jeho nejdůležitější součástí byla ovšem obrazovka, která měla stejnou velikost a orientaci jako tisknutá stránka s rozlišením 606 x 808 bodů. Umožňoval zapnutí a vypnutí jednotlivých pixelů, což byl značný pokrok oproti tehdejším terminálům, které zobrazovaly pouze textové znaky. Samotné grafické rozhraní ale nebylo příliš rozvinuté a podporovalo aplikace jako bitmapový grafický editor, správce souborů a textový editor. Výzkumníci si uvědomili, že je potřeba pro takovéto aplikace vytvořit nové prostředky pro jejich vývoj, a tak vznikl *Smalltalk*, první objektově orientovaný programovací jazyk.

V následujících letech se na rozvoji grafických uživatelských rozhraní podílely také další společnosti jako Apple, Microsoft a mnohé další. Není možné říci, že někdo vymyslel GUI, protože na něm spolupracovalo mnoho firem i jednotlivců a každý přispěl svou vlastní částí. Většina lidí v dnešní době považuje vývoj těchto rozhraní za stagnující, ne příliš se lišící od původních konceptů s plochou, myší, okny a ikonami. Je to mimo jiné tím, že uživatelé si na daný přístup již zvykli a vývojáři se obávají neúspěchu svých produktů, pokud by se od tohoto schématu odklonili. I když tedy

¹ http://www.invisiblerevolution.net/engelbart/full_62_paper_augm_hum_int.html

nová rozhraní obsahují stále stejné klíčové principy, stále se vytvářejí nové funkce a možnosti interakce člověka s počítačem.

2.2 3D uživatelská rozhraní

Jelikož žijeme v trojdimenzionálním světě, grafická rozhraní se začala vyvíjet tak, aby se co nejvíce přiblížila tomuto konceptu. Vznikla tedy 3D uživatelská rozhraní, která umožňují reprezentaci nebo manipulaci s 3D objekty. Některá se také snaží o zefektivnění uživatelské interakce. V zásadě je možné přístup k nim pojmut dvěma způsoby. Objekty lze buď zobrazovat na 2D zařízeních, jako jsou klasické monitory, nebo je možné pro jejich reprezentaci využít speciální technologie, které uživateli poskytuje prostorový vjem.

Důvodů k nástupu takovýchto 3D rozhraní je celá řada. Vývojáři se snaží přiblížit více reálnému světu, ať už je řeč o pomyslném listování ve fotoalbu nebo průjezdu městem v auto navigaci. Dalším důvodem je fakt, že takováto rozhraní jsou tak říkajíc „pěkná na pohled“, což zpřijemňuje práci a uživatelé mají pocit, že pracují s dokonalejším a modernějším produktem. Tato bude vycházet zejména ze zdroje [3]. Konkrétní příklady přístupů budou zmíněny v kapitole 2.3.

2.2.1 Historie

S nástupem nových technologií v oblasti 3D grafiky, virtuálního prostředí, rozšířené reality a nejrůznějších technologií simulátorů se začali rozvíjet i aplikace pro jejich využití. Objevili se dosud neprozkoumané výzvy a nápady v návrhu uživatelských rozhraní.

V roce 1968 americký vědec a nositel Turingovy ceny Ivan Sutherland sestrojil první displej, který byl umístěn na hlavě a sledoval její pohyby. Domníval se, že toto zařízení otevře zcela nové obzory. Předběhl ovšem svou dobu, a tak až na přelomu osmdesátých a devadesátých, kdy se objevili technologie jako 3D stereoskopická grafika, miniaturní CRT displeje, systémy mapující pozici a zařízení pro interakci (například datové rukavice), bylo možné vytvářet systémy virtuálního prostředí.

Ze začátku bylo virtuální prostředí používáno hlavně počítačovými inženýry a vědci pro jednoduché aplikace. Tyto aplikace nabízely vývojářům nové výzvy pro zdokonalování, nicméně co se týče uživatelské interakce, na tom nebyly nejlépe. Při vývoji komplexnějších aplikací ovšem výzkumníci zjistili, že spíše než technologie je jim překážkou nedostatek znalostí o návrhu rozhraní. Naštěstí v době, kdy se dostal vývoj virtuálního prostředí do krize, principy pro správný návrh rozhraní a uživatelskou interakci již byly v oblasti osobních počítačů relativně vyspělé.

Tyto znalosti napomohli ke zlepšení systémů virtuálního prostředí. Stále však existovali nezodpovězené otázky. Například jak bude uživatel pohybovat objekty? Kde bude ve 3D světě umístěno menu a jak z něj budou vybírány položky? Otázky jako tyto naznačovali, že bude potřeba vzniku nového odvětví uživatelské interakce zabývající se přímo návrhem rozhraní pro 3D prostředí. Ve stejnou dobu se rozvíjely i jiné technologie, jako 3D grafika pro stolní počítače nebo rozšířená realita, a vytváření jejich uživatelských rozhraní bylo pro vývojáře také značně problematické. Společné téma těchto technologií byla práce v 3D kontextu. Proto se nově vzniklá oblast uživatelské interakce nazývá návrh 3D uživatelských rozhraní nebo 3D interakce uživatele s počítačem.

2.2.2 Možnosti ovládání

V této kapitole budou popsány některé z mnoha způsobů ovládání 3D uživatelských rozhraní.

Desktopová vstupní zařízení

V prvé řadě se jedná o klasický způsob pomocí myši, klávesnice, joysticku a podobně. Ten s sebou nese jisté výhody i nevýhody. Nespornou výhodou je pochopitelně cena a dostupnost těchto zařízení. Pokud je řeč o zobrazení na klasickém monitoru, je tento přístup velice efektivní a přirozený. Na druhou stranu v prostoru je tento druh ovládání nedostačující, jelikož kupříkladu myš nabízí pohyb pouze ve dvou osách a neumožňuje tak manipulaci s 3D objekty v tom pravém slova smyslu. Klávesnice pak nabízí například pohyb po scéně pomocí šipek a je hlavně určena na zadávání alfanumerických dat. V tomto případě může nastat problém, pokud uživatel používá displej umístěný na hlavě a tudíž nemůže klávesnici využívat. Místo ní lze použít zařízení pro symbolický vstup, které se drží v ruce a stisknutím kombinací kláves lze zadat alfanumerické znaky.

Zařízení sledující pohyb

Dalším způsobem jsou zařízení, která sledují pohyb. Patří sem mimo jiné různé druhy datových rukavic. Třeba rukavice *P5 Glove*¹ (viz Obrázek 2.1a) pomocí mapování polohy ruky vůči snímači umístěnému před uživatelem a ohýbání prstů umožňuje nejrozličnější využití ovládání 3D scény. Jiné dokonce využívají mechanicky řešenou zpětnou vazbu, která poskytuje realistický vjem uchopování předmětů, tedy vnímání jejich tvaru, váhy a podobně. Při tomto snímání pohybu nemusí být ovšem snímána jen ruka, ale dokonce celé tělo (viz Obrázek 2.1b). Tímto způsobem však není nutné ovládat pouze počítač, ale je velmi využíván například při vývoji her, filmů a nejrozličnějších simulacích, kde je zapotřebí přesně vystihnout pohyb postavy. Takto zaznamenávat pohyb je možné buď pomocí senzorů umístěných přímo na rukavici či obleku, nebo lze objekty snímat kamerou a výsledný pohyb vypočítat z polohy klíčových bodů v obraze. Objektem je zde myšleno jak konkrétní zařízení (rukavice), tak přímo části lidského těla nebo v obraze dobře rozlišitelné běžné předměty. Je tak možné se například „rozhlížet“ po scéně pohybem hlavy do stran nebo pomocí gest ovládat hudební přehrávač či hru. Tohoto principu využívá například firma Microsoft se senzorem *Kinect*.

Jiná zařízení sledují pohyb očí, magnetických nebo akustických podnětů. Sledování pohybu očí lze přitom využít nejen pro přímé ovládání rozhraní, ale také u psychofyzických experimentů, které vyhodnocováním pohledu uživatele pomáhají zlepšovat návrh rozhraní.



Obrázek 2.1a – P5 Glove

Převzato z <http://www.cwonline.com>



Obrázek 2.1b – Motion capture oblek

Převzato z <http://www.xsens.com/>

¹ <http://www.cwonline.com/>

Lidské tělo jako vstupní zařízení

Velice mocným přístupem k ovládání 3D aplikací je získávání signálů přímo z lidského těla. Uživatel se tak přímo stává vstupním zařízením. Lze tak kupříkladu zadávat povely hlasem. Tato technika je užitečná zejména v případě, kdy má uživatel zaměstnány obě ruce, přináší však také mnoho problémů. Jestliže mikrofon není umístěn přímo u hlavy uživatele, ale snímá zvuk z širšího prostoru, může zaznamenávat nechtěné zvuky prostředí. Další komplikací je určení, kdy aplikaci uživatel zadává příkazy a kdy se jen baví s kolegy. Využívá se k tomu takzvané řešení *push-to-talk*, tedy že aplikace vyhodnocuje hlasové příkazy, pouze pokud je stisknutá nějaká klávesa nebo při jiném podnětu.

Dalším zajímavým přístupem je ovládání pomocí signálů vysílaných mozkem. Ty je možné získat z elektroencefalogramu (EEG), přičemž má uživatel nasazenu pouze čelenku nebo čepici, která tyto signály snímá. Poněkud invazivnější způsob je chirurgické implantování mikroelektrod do mozku do oblasti motoriky. Tento přístup ovšem zatím není možné prakticky použít. Příkladem komerčně dostupného zařízení, které kromě mozkové aktivity monitoruje také pohyb očí a obličejových svalů, je čelenka s názvem *Nia*¹ (viz Obrázek 2.2). Ta by po kalibraci s konkrétním uživatelem měla nahradit klávesnici například v počítačových hrách nebo posloužit tělesně postiženým lidem, kteří nemají jinou možnost interakce s počítačem.

Výzkumné centrum NASA také vytvořilo zařízení, které čte bioelektrické svalové signály z předloktí. To pak dokáže z definovaných vzorů vybrat ten správný a podle něj provést nějaký konkrétní úkon s rozhraním.



Obrázek 2.2 – Nia čelenka

Převzato z <http://www.ocztechnology.com/>

3D myši

3D myši jsou podobné klasickým myším, na rozdíl od nich se ovšem nepohybují jen po plochém povrchu, ale je možné je uchopit a fyzicky s nimi pohybovat v 3D prostoru k získání informace o pohybu nebo orientaci. Jednu s prvních 3D myší představil Colin Ware v roce 1988 a nazval ji příhodně „netopýr“. Tato zařízení mohou být kromě prostého držení v ruce umístěna přímo na ní, obvykle na ukazováku, a tlačítka jsou pak stisknuta palcem. Problémem je však omezený počet těchto tlačítek.

Využití všech těchto přístupů je takřka neomezené a nevztahuje se pouze k ovládání 3D uživatelských rozhraní. Vyžaduje ovšem v jistých případech určitou zručnost a cvik, což může být pro některé uživatele zvyklé na klasická zařízení poněkud problematické.

¹ <http://www.ocztechnology.com/>

2.2.3 Výstupní zařízení

Jak již bylo řečeno, nezbytnou komponentou 3D uživatelského rozhraní je hardware, který prezentuje informace uživateli. Tyto informace mohou být vizuálního, sluchového nebo hmatového charakteru. Nejobvyklejší formou prezentace je bezesporu forma vizuální, tedy prostřednictvím displeje.

Takový displej, pokud má sloužit k zobrazování 3D rozhraní, má následující charakteristiky:

- *Rozsah zobrazení a rozsah pohledu*
Rozsah zobrazení určuje prostor, na který je uživateli promítán obraz. Měří se ve stupních úhlu pohledu, tedy pokud se jedná například o válcovitý displej, ve kterém stojí uživatel, má displej rozsah 360° horizontálně. S tím souvisí termín rozsah pohledu, který odkazuje na úhel, který může uživatel v jednom okamžiku vidět na displeji.
- *Rozlišení*
Měří se v DPI (bodů na palec) a závisí na počtu pixelů a velikosti displeje.
- *Geometrie obrazovky*
Displeje mají také různé tvary, například obdélníkové, ve tvaru L, polokulovité a podobně. Obrazovky neobdélníkovitého tvaru obvykle vyžadují speciální zobrazovací algoritmy.
- *Přenos světla*
Charakterizuje způsob, jakým je světlo skutečně přenášeno na povrch displeje. Těchto způsobů existuje několik, včetně přední projekce, zadní projekce, laserem přímo na sítnici nebo užití speciální optiky.
- *Obnovovací frekvence*
Určuje rychlost v hertzech (Hz) s jakou je obraz na displeji obnovován. Tento parametr je u 3D zobrazení velice důležitý.
- *Ergonomie*
Je důležité, aby se uživateli s displejem pohodlně pracovalo, obzvláště pokud se jedná o displej umístěný na hlavě.

Dále budou zmíněny některé konkrétní druhy displejů a jejich výhody a nevýhody s ohledem na 3D interakci.

Monitory

Běžné monitory jsou v dnešní době vcelku levnou záležitostí, a proto jsou využívány v celé řadě 3D aplikací, například modelování, vědecké vizualizace a počítačové hry. S přídatným hardwarem, jako jsou stereo brýle, mohou být také použity k dosažení prostorového vidění. Takové monitory by potom měli mít obnovovací frekvenci minimálně 100 Hz, jelikož je obraz produkován pro každé oko zvlášť, což prakticky snižuje výslednou frekvenci na polovinu. Z tohoto důvodu se pro tento účel klasické LCD monitory nehodí, protože dosahují frekvence zhruba jen 85 Hz.

Co se týče stereo brýlí, rozdělují se na aktivní a pasivní. Aktivní brýle střídavě uzavírají clony před levým a pravým okem ve stejné frekvenci, na jaké pracuje monitor, který střídavě vykresluje obrazy pro každé oko zvlášť. Clony se uzavírají tak rychle, že to uživatel ani nevnímá a má pocit, že vidí prostorově. Pasivní brýle se potom dělí na polarizační nebo spektrální. Polarizační filtrují dva oddělené překryté obrazy opačně polarizovanými filtry. Spektrální brýle mají dva filtry různé barvy, nejčastěji modré a červené. Dva překryté obrazy jsou pak také obarveny těmito barvami, každé oko pak vidí jen jeden obraz v barvě shodné s filtrem.

Všeobecně poskytují aktivní brýle větší kvalitu prostorového vidění, ovšem na druhou stranu pasivní brýle jsou cenově dostupnější a odpadají problémy s jejich synchronizací s frekvencí monitoru.

Displeje obklopující uživatele

Tyto displeje (angl. *surround-screen displays*) jsou tvořeny většinou třemi a více projekčními stěnami, mezi kterými stojí uživatel. Zpravidla je obraz promítán na stěny zezadu, aby uživatel nevrhal stín. Mezi výhody jejich použití patří vysoké rozlišení, velký rozsah zobrazení a velký rozsah pohledu, jelikož je využíváno i periferního vidění. Je u nich také možno používat aktivní a pasivní stereo brýle. Rovněž lze v tomto prostředí spojit reálné předměty s virtuálními, například uživatel může sedět v replice automobilu a pohybovat se po virtuální silnici. Jednou z největších nevýhod je vysoká cena těchto zařízení a fakt, že zabírají velký prostor.

První takový displej pro virtuální realitu byl vytvořen v Electronic Visualization Laboratory na univerzitě v Chicagu v roce 1992. Tento systém byl nazván *CAVE* a skládal se ze tří stěn a podlahy a používaly se při něm aktivní stereo brýle.

Polokulovité displeje

Dalším typem displejů založených na projekci jsou displeje polokulovitého tvaru. Jak již název napovídá, umožňují promítání obrazu v rozsahu 180°. K tomu je využito speciální optiky projektoru a softwaru, který správně namapuje obraz. Uživatel pak sedí naproti displeji za malým stolem a interaguje rozhraním prostřednictvím myši, klávesnice nebo některým s 3D vstupních zařízení. Jelikož se většinou využívá přední projekce, obraz bývá jasnější než u promítání zezadu.

Displeje umístěné na hlavě

Displeje, které má uživatel umístěn přímo na hlavě (angl. *head-mounted displays* HMD) jsou velice komplexní zařízení obsahující elektronické, optické, mechanické a audio komponenty. Obraz je pak zobrazován přímo před oči za použitím jedné nebo dvou LCD nebo CRT obrazovek. Výhodou těchto displejů je uživatelova volnost v pohledu, jelikož se obraz přizpůsobuje otáčení jeho hlavy a tím poskytuje rozhled 360°. Ačkoliv je většinou zobrazován počítačem generovaný obraz, některé displeje snímají reálný svět externí kamerou a do něj pak přidávají virtuální objekty. Tyto technologie se pak používají zejména v rozšířené realitě.



Obrázek 2.3 – Displej umístěný na hlavě

Převzato z <http://www.vrealities.com>

2.3 Existující přístupy k 3D UI

V této kapitole budou zmíněny některé konkrétní přístupy k 3D uživatelským rozhraním, jejich přednosti a nedostatky. Jelikož předmětem této práce je zobrazování souborových struktur, bude se jednat vesměs o příklady rozhraní s ním spojených.

2.3.1 Cooliris

Rozhraní s názvem *Cooliris*¹ umožňuje prohlížení nejrůznějších obrázků a videí přímo v internetovém prohlížeči. Jeho nespornou výhodou je tedy dostupnost pro běžné uživatele, kteří si jej mohou zdarma instalovat do svého oblíbeného prohlížeče. Cooliris může být použit pro zobrazení jakýchkoliv obrázků, ať už například výsledků vyhledávání na *www.google.com* nebo při procházení fotek přátel na *www.facebook.com*.



Obrázek 2.4 – Rozhraní Cooliris

Převzato z <http://www.cooliris.com/>

Základem rozhraní je „zed“ náhledů obrázků, které jsou načítány z právě vybraného zdroje. Tyto takzvané kanály třídí obrázky do několika kategorií podle umístění nebo zaměření. Tato nabídka již ovšem postrádá jakýkoliv 3D efekt. Po kliknutí na náhled se obrázek zvětší, a pokud například pochází z nějakého článku, zobrazí se i úryvek textu, ve kterém se vyskytuje. Po zdi je možné se pohybovat dvěma způsoby. Prvním z nich je pomocí posuvné lišty, přičemž je délka jejího posunutí závislá na rychlosti pohybu po zdi. Druhou možností je kliknutím a tažením myši do stran opět s proměnlivou rychlostí. Tento způsob je efektivní, nicméně při využívání této metody snadno dochází ke zvětšování obrázků po nechtěném kliknutí na ně.

Co se tedy týče 3D vjemu, ten je zde tedy omezen na „vysouvání“ obrázků k uživateli a na „rozhlížení se“ podél galerie. Celkově však Cooliris nabízí vcelku zajímavé řešení prostorového procházení multimédií.

¹ <http://www.cooliris.com/>

2.3.2 BumpTop 3D Desktop

Neméně zajímavým využitím 3D rozhraní je *BumTop 3D Desktop*¹, které, jak již název napovídá, slouží jako náhrada klasické plochy. Vychází ze skutečnosti, že v reálném světě jsou předměty, jako knihy, CD a fotografie, organizovány například na stole ve formě „hromádek“ a poskytují možnost okamžitě získat potřebné informace. Stejně tak jsou tedy organizovány objekty v této virtuální ploše. Ta je reprezentována jako místnost, po které je možné se volně pohybovat. S jednotlivými soubory lze potom zacházet jako s reálnými objekty, tedy dodržují fyzikální zákony. Je možné jimi hodit, zmačkat je nebo je ohnout.

Jak již bylo řečeno, hlavní používanou strukturou je stoh, lze však také soubory zanechat neuspořádané. Stoh lze pak procházet mnoha způsoby, například obracet stránky jako v knize nebo rozložit soubory do mřížky. Umožňuje také jejich řazení, změnu velikosti a připínání na stěny virtuální místnosti.

Tento způsob zprávy plochy je velice inovativní a atraktivní, což dokazuje také odkoupení tohoto (původně freeware) projektu společností Google. Za zmínku také stojí podpora multi-touch, tedy interakce pomocí přímého dotyku displeje několika prsty současně, která přispívá k celkové intuitivnosti rozhraní a díky současnému trendu dotykových displejů je jeví jako velmi perspektivní.



Obrázek 2.5 – Rozhraní BumTop 3D Desktop

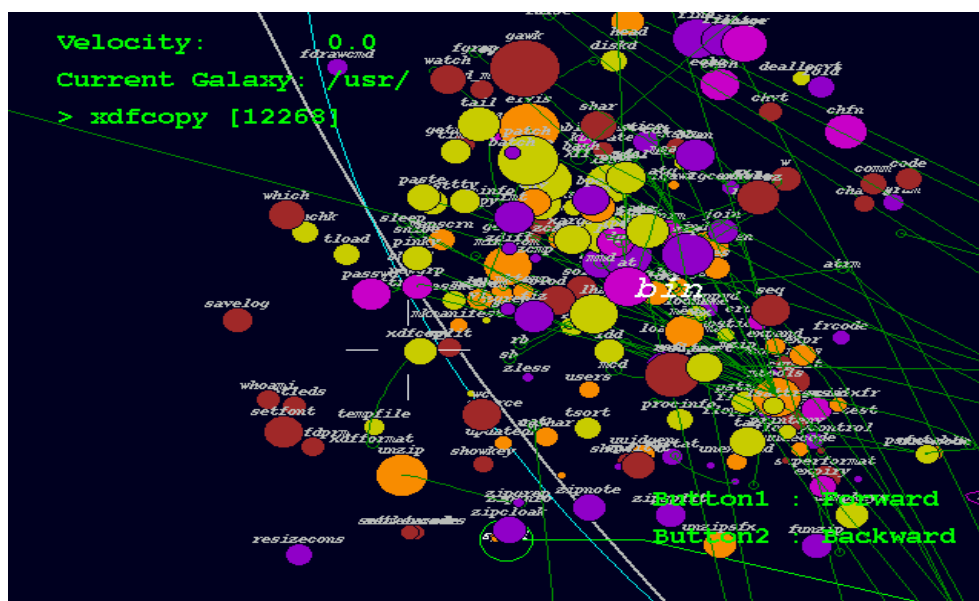
Převzato z <http://www.strangesystems.com>

¹ <http://bumptop.com/>

2.3.3 Prohlížeče souborových systémů

Co se týče procházení souborového systému ve 3D, existuje několik přístupů. Některé jsou pojaty formou reálného světa, tedy že uživatel prochází místností a jednotlivé sekce souborů jsou reprezentovány „virtuálními stánky“, ve kterých jsou na stěnách umístěny zástupci aplikací a podobně. Příkladem jsou produkty společnosti 3DNA¹. Tento přístup je sice zajímavý, ovšem vcelku časově neefektivní, jelikož prostor, ve kterém se takové soubory nacházejí, může být často rozsáhlý a přesuny mezi sekcemi zabírají zbytečně čas.

Jiná rozhraní se snaží zachytit spíše hierarchii souborů, což vede většinou spíše k jistému chaosu, než k inovativnosti a efektivitě při plném využití trojrozměrného prostoru. Některá takováto rozhraní jsou například XCruiser², Tactile3D³ nebo FSV⁴.



Obrázek 2.6 – Rozhraní XCruiser
Převzato z <http://xcruiser.sourceforge.net/>

Hlavní nevýhodou těchto přístupů je skutečnost, že se používají zpravidla na běžných stolních počítačích s použitím klávesnice a myši jako vstupního zařízení a klasického monitoru jako výstupního zařízení. Tedy dochází k zobrazování 3D prostředí na 2D zařízení. To má za následek nedostatečný vjem třetího prostoru a špatný přehled o scéně. Také standardní myš neumožňuje přirozený pohyb ve 3D prostředí.

Z těchto důvodů se tato rozhraní zatím v praxi příliš nevyužívají, nicméně s nástupem nových technologií pro trojdimenzionální zobrazení se budou stále více uplatňovat.

¹ <http://3dna.net>

² <http://xcruiser.sourceforge.net/>

³ <http://www.tactile3d.com>

⁴ <http://fsv.sourceforge.net/>

2.3.4 Blízká „budoucnost“

Do jisté míry revolučním produktem (i když ne přímo souvisejícím s hierarchickými strukturami) je *Cheoptics360* od firmy viZoo¹. Ten nabízí vjem prostorových objektů zobrazených v jakési obrácené pyramidě. Je určen převážně pro prezentaci nejrůznějších produktů, ale jeho možnosti jsou takřka neomezené. Neumožňuje ovšem přímou interakci uživatele se zobrazovanými objekty.

Na první pohled se může 3D projekce „jen tak do vzduchu“ jevit jako nereálná, *Cheoptics360* však funguje na vcelku prostém principu. Na každou stěnu pyramidy promítá projektor několikrát odražený obraz. Stěny jsou tvořené z průhledného materiálu s lehkým černým tónem, který umožňuje samotné zobrazení. Spojením těchto čtyř projekcí objekty jakoby plují ve vzduchu. Tvůrci uvádějí, že zobrazování je možné za jakýchkoli světelných podmínek, ale samozřejmě jelikož jsou video sekvence promítány na téměř průhledný povrch, jsou i samotné objekty poloprůhledné. Tento produkt je nabízen v několika velikostech a variantách. Maximální rozměry obrazu jsou 140 x 95 cm, což umožňuje skutečně detailní prezentaci například automobilu. Mohl by v budoucnu také nahradit běžnou projekci v kinech a své uplatnění by našel i v herním průmyslu. Čerpáno bylo z [6].



Obrázek 2.7 – Cheoptics360
Převzato z <http://www.portlandadfed.com>

Obecně jsou tedy 3D uživatelská rozhraní na vzestupu a budeme se s nimi setkávat čím dál tím více. Bude potřeba k tomu ale přizpůsobit nejen softwarovou část, ale i hardware. Otázkou ovšem je, zda v některých případech není skutečně rozumnější zůstat u klasického 2D zobrazení, které sice neposkytuje tak dokonalý vjem reálného světa, ale disponuje svou přehledností a do značné míry také efektivitou, to ovšem samozřejmě závisí na konkrétním případě rozhraní.

¹ <http://www.vizoo.com/>

3 OpenSceneGraph

OpenSceneGraph je multi-platformní grafický soubor nástrojů určený pro vývoj aplikací z oblasti počítačových her, simulátorů, vědeckých vizualizací a virtuální reality. Konkrétně je využíván organizacemi, jako jsou Boeing, NASA, Magic Earth, Delta 3D a dalšími. Jedná se o open source aplikaci, proto je vhodná pro komerční i nekomerční využití. Je založena na principu grafu scény, který bude vysvětlen dále, a poskytuje objektově orientovanou nastavbu nad OpenGL. To umožňuje rychlejší a pohodlnější vývoj aplikací, jelikož se vývojáři nemusí zabývat implementací a optimalizací nízkourovňových volání grafických funkcí. Je psána ve standardním C++ a OpenGL. Tato kapitola vychází zejména ze zdrojů [1], [2] a [4].

Mezi hlavní přednosti OpenSceneGraph patří:

- *Výkonnost*
Podporuje různé typy ořezávání (culling), úroveň detailu (LOD), řazení OpenGL stavů, pole vrcholů, OpenGL shadery a mnoho dalšího. Veškeré funkce jsou optimalizované a tím činí OpenSceneGraph velmi výkonným grafickým nástrojem.
- *Přenositelnost*
Jádro aplikace bylo navrženo tak, aby nebylo závislé na konkrétní platformě. To umožnilo rozšíření z původní platformy IRIX na Linux, FreeBSD, Windows, Mac OSX, Solaris, HP-UX, AIX a dokonce i na PlayStation2. Součástí jádra je i knihovna osgViewer, která dovoluje integraci s nástroji, jako jsou Qt, GLUT, MFC a další.
- *Rozšiřitelnost*
OpenSceneGraph je vhodný pro psaní vlastních rozšiřujících knihoven a zásuvných modulů a také umožňuje využití několika grafických kontextů a více vláken.
- *Podpora*
Díky aktivní komunitě vývojářů se OSG neustále rozvíjí a podporuje tak nejnovější OpenGL rozšíření a grafické techniky.

V OSG je velmi důležitá správa paměti, aby se zabránilo jejím únikům a neočekávanému chování programu. Základ správy paměti sestává z alokace, tedy poskytnutí paměti objektu vyhrazením potřebných paměťových bloků, a dealokace, což znamená uvolnění paměti pro další využití jinými objekty, jakmile není dále využívána.

Některé programovací jazyky jako C#, Java nebo Visual Basic používají tzv. *garbage collector*, které uchovávají počet objektů odkazujících se na blok paměti a v případě, že jejich počet dosáhne nuly, tento blok uvolní. Standardní C++ tento přístup bohužel nepodporuje.

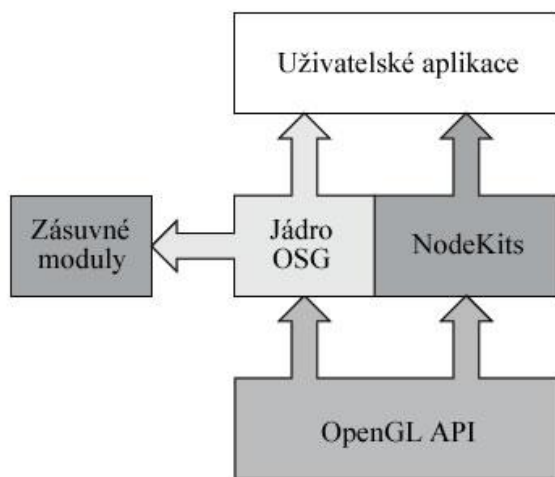
Naštěstí OSG poskytuje tzv. *smart pointer*, který plní funkci garbage collectoru. Poskytuje v podstatě stejnou funkčnost jako klasický ukazatel, nicméně zajišťuje také automatické inkrementování a dekrementování počtu referencí na objekt. Pokud je tedy například objekt alokován na haldě (pomocí `new`), bez použití smart pointeru jako lokální proměnná nějaké funkce, zůstane po jejím opuštění paměť alokována až do konce běhu programu. Naproti tomu při použití smart pointeru je po opuštění funkce zavolán jeho destruktorka, který dekrementuje počet referencí, a jestliže je roven nule, je paměť korektně uvolněna. Umožňuje tedy efektivní řízení paměti, objekty jsou vymazány, jakmile nejsou využívány, což poskytuje více paměti pro aplikace s omezenými zdroji. Dále při jejich použití dochází méně k chybám, smart pointery automaticky inicializují a odstraňují ukazatele. Nemohou tedy existovat žádné ukazatele odkazující do nekorektního bloku paměti. A v neposlední

řadě usnadňují hledání případných chyb, kdy lze snadno zjistit počet referencí na objekt a další užitečné informace.

3.1 Graf scény

Graf scény je stromová datová struktura, která definuje prostorové a logické uspořádání grafické scény pro zajištění efektivního řízení a renderování grafických dat. Skládá se z tzv. uzlů, přesněji z jednoho kořenového uzlu (*root node*), skupinových uzlů (*group nodes*), které mohou obsahovat jakýkoliv počet uzlů potomků (*child nodes*), a z listových uzlů (*leaf nodes*), které už nemají žádné další potomky. Sdružování uzlů potomků pod skupinovým uzlem slouží ke sdílení informací o jejich rodiči a je s nimi možné zacházet jako s jedním objektem. Pokud není nastaveno jinak, veškeré operace prováděné s rodičovským uzlem se provedou i na jeho potomcích.

Kromě OpenSceneGraphu využívají koncept grafu scény i jiné aplikace, například Maya, AutoCAD, CorelDraw a další.



Obrázek 3.1 – Architektura OSG
(Převzato ze zdroje [2])

3.2 Komponenty

Jádro OpenSceneGraphu je tvořeno těmito třemi základními knihovnami:

- *Knihovna osg*
Poskytuje základní elementy k tvorbě grafu scény, jako jsou uzly, geometrie, renderovací stavy, textury a metody s nimi spojené. Také obsahuje některé matematické třídy, které jsou nezbytné pro operace s vektory a maticemi.
- *Knihovna osgDB*
Zajišťuje podporu čtení a zápisu 2D a 3D souborů (.3ds, .lwo, .flt, .jpg, .tiff). Pro spravování velkých kolekcí dat umožňuje dynamické načítání a uvolňování segmentů grafu scény.
- *Knihovna osgUtil*
Tato knihovna zajišťuje konečnou renderovací fázi, ve které prochází stromem, ořezává jednotlivé grafické rámce a nakonec převádí OpenSceneGraph scénu na sérii OpenGL příkazů. Poskytuje také algoritmy pro modifikaci polygonů.

Kromě těchto základních knihoven obsahuje i mnoho přídavných, tzv. *NodeKits*:

- *Knihovna osgGA*
Poskytuje rozhraní pro korektní zobrazení scény nezávisle na okenním systému, na kterém aplikace běží. Také pomáhá s obsluhou uživatelských událostí způsobených periferními zařízeními, například klávesnicí nebo myší.
- *Knihovna osgViewer*
Definuje třídy spojené se samotným zobrazením na systémech Win32, X11, Carbon atd.
- *Knihovna osgText*
Tato knihovna plně podporuje fonty založené na FreeType projektu a může tak do scény vykreslit 2D nebo 3D text.
- *Knihovna osgParticle*
Umožňuje efekty založené na částicích, tedy například oheň, kouř, exploze a další.
- *Knihovna osgShadow*
Poskytuje různé techniky pro renderování stínů.
- a mnohé další

4 Návrh rozhraní

V této kapitole budou zmíněny jednotlivé části návrhu aplikace, počínaje volbou vhodného rozhraní a dále pak některé konkrétní návrhy.

Záměrem bylo tedy vytvořit uživatelské rozhraní pro zobrazování hierarchických struktur, které by využilo třetího rozměru netradičním řešením klasického dvojdimenzionálního přístupu. Pro demonstraci takovýchto struktur byl zvolen souborový systém, který je stromově uspořádán. Z důvodu dostupnosti běžným uživatelům bylo zvoleno zobrazení na klasickém monitoru a ovládání aplikace myší.

4.1 Volba vhodného rozhraní

V první řadě je potřeba určit charakter 3D uživatelského rozhraní. Je nutné si uvědomit, že jeho primárním účelem je zpřístupnit funkce nějaké konkrétní aplikace pomocí grafických prvků. Mělo by být proto přehledné, efektivní a do jisté míry inovativní.

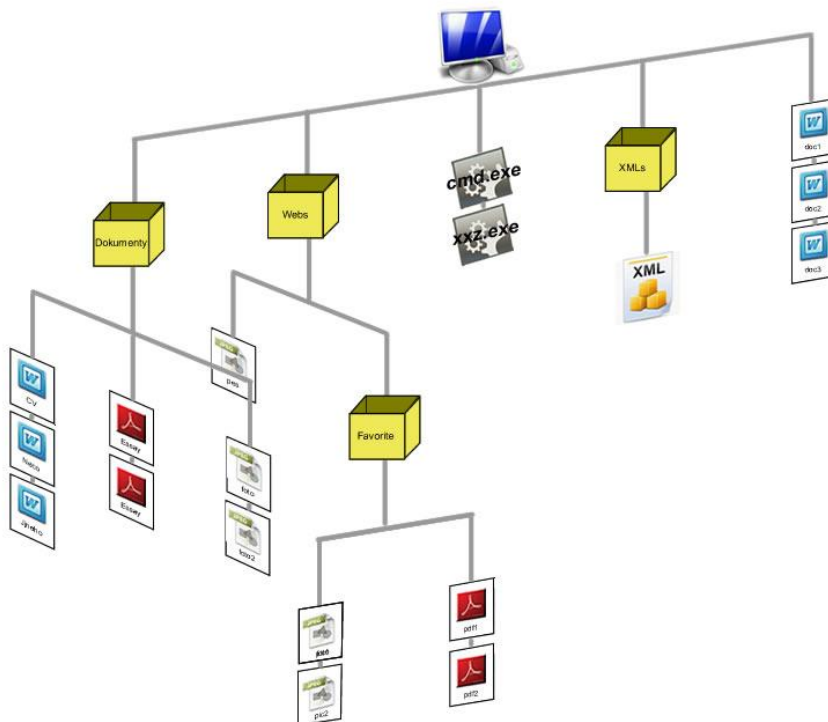
Přehlednost je klíčovou vlastností každého rozhraní, nicméně obzvláště u jeho trojdimenzionálního zobrazení je třeba na ni klást velký důraz. Třetí rozměr zobrazený na 2D zařízení totiž může uživatele zmást a místo výhody se z něj stává nevýhoda. Proto je potřeba zajistit, aby veškeré důležité prvky rozhraní, jako jsou například nejruznější menu a podobně, byly co nejlépe viditelné. To většinou znamená jejich umístění ve scéně co nejbližší pozorovateli, aby bylo hned patrné, že mají velký význam a aby se nestalo, že budou kupříkladu částečně nebo zcela zakryté jiným objektem. Volba barevného schématu hraje také důležitou roli, protože vhodné uskupení barev silně přispívá k celkové přehlednosti rozhraní, a samozřejmě práce s ním by měla být pohodlná a příjemná.

Efektivnost jde ruku v ruce s přehledností. Je-li rozhraní přehledné, uživatel se v něm snadno zorientuje a dokáže rychle provádět úkony, které zamýšlel. U 3D rozhraní je důležité, obzvláště pokud je ovládáno klasickými zařízeními jako je myš a klávesnice, aby byly pro uživatele jasné a snadné úkony spojené s třetím rozměrem. Tedy například přílišná volnost v pohybu v prostoru je spíše na škodu, jelikož pozorovatel mnohdy ztrácí pojem o aktuální pozici v rámci celé scény. Je proto vhodnější nabídnout jen několik předdefinovaných pohledů na scénu, mezi kterými je schopen se snadno pohybovat. Uživatel se tak může zaměřit spíše na samotné funkce rozhraní, než na jeho ovládání.

Inovativnost je velice choulostivý prvek rozhraní. Pokud se pokusíte udělat je příliš inovativní, může se stát nepřehledným a tudíž neefektivním. Uživatelé, zvyklí na svůj zavedený přístup k jistým úkonům, se ho neradi vzdávají a je tedy vhodné zvolit nějaký kompromis mezi tradičním a vlastním novým přístupem. Samozřejmě jestliže se jedná o opravdu revoluční rozhraní, není třeba se omezovat. Všechny tyto charakteristiky byly při návrhu rozhraní pro tuto práci vzaty v úvahu. Vzniklo tak několik variant, přičemž některé z nich budou dále detailněji popsány.

4.2 Stromová struktura prohlížeče

Původně bylo zamýšleno demonstrovat hierarchii souborů v podobě jakéhosi stromu, ve kterém by složky byly představovány boxy a jejich potomky a zároveň tedy podstromy by byly soubory a další složky v ní obsažené. Soubory by byly řazeny do jednotlivých větví podle jejich typu a názvu. Každá vrstva stromu by byla otočená o 90° vůči předchozí, z důvodu omezení střetů objektů vedlejších větví. Mezi hlavní výhody toho rozhraní by patřila možnost přesouvání celých podstromů na libovolné místo ve stromu. V tomto bodě by tedy byl značný rozdíl oproti běžným prohlížečům, které pro snadné přesouvání složek buď nutí uživatele otevřením další instance programu, nebo nabízí současné zobrazení dvou umístění. Pokud by však byla zobrazena celá (námi prozkoumávaná část) stromu, měl by uživatel kompletní přehled o struktuře, a tak snadno a rychle přesouval soubory. Další funkcí spojenou s přesouváním by byla schránka, do které by bylo možné „zavěsit“ několik podstromů pro pozdější přesunutí. Uživatel by do ní mohl takto uložit soubory a složky, přesunout se do kýžené části stromu a soubory tam přemístit.



Obrázek 4.1 – Návrh stromové struktury prohlížeče médií

Již v počátku implementace se ovšem objevily problémy, kterým do té doby nebyla věnována taková pozornost. Ukázalo se, že pokud by rozhraní mělo být použito pro rozsáhlé souborové struktury, řádově desítky souborů a složek v každé úrovni, i s přihlédnutím na otočení jednotlivých vrstev by objekty spolu kolidovaly. Zamezit by se tomu dalo například skrytím (zavřením) vedlejších podstromů, čímž by se ovšem ztratila podstata toho rozhraní, tedy širší přehled o zanoření souborů.

Dalším závažným problémem se zdál být pohyb pohledu na tak rozsáhlé struktury. Jak bylo zmíněno v předchozí kapitole, pokud by měl uživatel plnou volnost pohybu kamery, těžko by se za použití klasických vstupních zařízení zorientoval a „nasměroval“ pohled kamery na požadovaný objekt. Naproti tomu, jestliže by se při otevření příslušné složky kupříkladu kamera zaměřila na první nově otevřený soubor a umožňovala postupné procházení dalších, ztratil by uživatel přehled o zbytku stromu a musel by složitě procházet stovky souborů do hloubky scény.

Z těchto důvodů bylo od tohoto konceptu upuštěno a bylo potřeba vymyslet praktičtější, pro uživatele pohodlnější rozhraní. Takové, které by poskytovalo informace o zanoření a struktuře souborového systému a umožňovalo také přehledné procházení složek obsahujících vysoký počet souborů a dalších složek.

4.3 Prohlížeč formou zdi souborů

Použitelným návrhem se nakonec ukázalo vytvoření jakési „nekonečné“ zdi, ve které se budou v několika řadách zobrazovat jednotlivé soubory z aktuálně vybrané složky. Tyto objekty pak kromě svého názvu včetně přípony budou obsahovat obrázek typu souboru. Tato zeď bude zabírat většinu obrazovky, resp. její vrchní část.

Ve spodní části by se pak mělo nacházet menu pro průchod složkami. To bude reprezentováno karuselem, ve kterém budou zobrazeny názvy podsložek aktuálně vybrané složky. Karuselem bude možné rotovat, aby bylo možné vybrat jakoukoli jeho položku. Po jejím výběru se stávající karusel posune o něco níž a na jeho místě se objeví nový s obsahem aktuálně vybrané položky. Takto bude možné se vcelku neomezeně zanořovat, přičemž si uživatel bude stále udržovat přehled o aktuální pozici v celkové struktuře.

Pro demonstraci prohlížení souborů, bylo pro tuto práci plánováno získávání skutečných dat o souborové struktuře z počítače, což pomůže odhalit některé chyby, které bez nasazení do skutečného prostředí nemusí být patrné.

4.3.1 Ovládání kamery ve scéně

U tohoto rozhraní jsme se rozhodli, jak bylo popsáno v předchozí kapitole, neumožnit uživateli naprosto volný pohyb s kamerou, nýbrž nabídnout mu několik pohledů, mezi kterými by mohl plynule přecházet.

Jelikož je záměrem, aby bylo rozhraní přístupné co nejširšímu spektru uživatelů, byla jako jediné (alternativně by bylo možné pro nějaké úkony použít klávesnici) vstupní zařízení pro manipulaci s rozhraním vybrána klasická myš. Kamera by se měla pohybovat v podstatě čtyřmi směry. Pohybem kurzoru myši k levému, resp. pravému okraji obrazovky se kamera otočí tímto směrem a zároveň se kamera posune podél zdi. Tímto „rozhlížením“ se tedy bude dát procházet jednotlivými položkami tvořícími zeď. Pokud se posune kurzor nahoru, kamera se přiblíží k souborům na zdi a umožní tak jejich pomalejší procházení s využitím opět levého nebo pravého okraje obrazovky. Tentokrát se ale nebude natáčet do stran, aby se mohl uživatel zaměřit na samotné soubory. Při umístění kurzoru do spodní části obrazovky se kamera posouvá dolů ke karuselům, aby bylo možné vidět několik zanoření doposud zobrazených složek souborového systému.

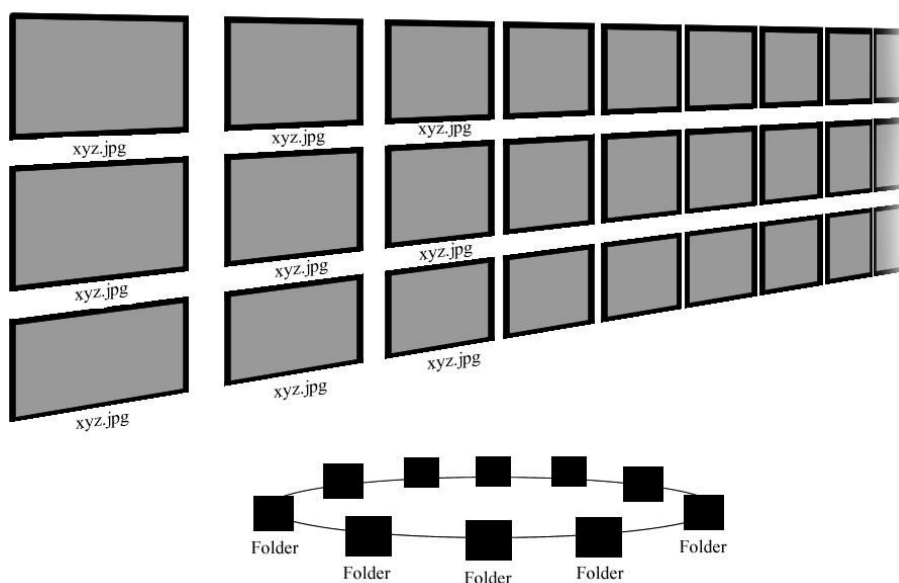
Ovládání pohledu by bylo možné, mimo jiné způsoby, provádět také pohybem hlavy. To ovšem od uživatele vyžaduje jednak alespoň webovou kameru, která by jej zabírala, potom také určitý cvik. Pokud by totiž při práci dělal například ještě další věc a často odhlížel od monitoru, aplikace by tyto pohyby interpretovala jako interakci s uživatelským rozhraním.

4.3.2 Detaily souborů

Vzhledem k tomu, že objekty představující soubory na zdi jsou kvůli jejich velkému počtu vcelku malé, je vhodné, aby měl uživatel možnost podívat se na detailnější informace o daném souboru, či přímo jejich náhledy. Jelikož by aplikace mohla být použita i pro jiné účely, zdá se užitečné mít možnost zobrazit hned dva náhledy současně. Například pokud by aplikace umožňovala úpravu fotografií, kdy by v levém náhledu byla fotografie původní a v pravém upravovaná, nebo kdyby byla použita jako internetový obchod, mohl by zákazník porovnávat parametry dvou produktů.

4.3.3 Jiné možnosti zobrazení

Vytvoření rovné zdi souborů bylo zvoleno z důvodu efektivního využití místa na obrazovce a přehlednému zobrazení potenciálně mnoha souborů. Pro navrhované rozhraní se jevily jako ideální tři řady souborů, bylo by ale možné zobrazit jich méně či více. Při nižším počtu by ovšem byl uživatel se příliš posouvat po zdi, naopak při více řadách by mohly být při určitém rozlišení obrazovky například názvy souborů špatně čitelné. Také tvar zdi by se mohl lišit, není problém, aby byla ve tvaru válce nebo oblouku. Tyto tvary však nepřinášejí žádnou výraznou výhodu a kupříkladu efekt válce při zobrazení velkého počtu souborů zaniká.



Obrázek 4.2 – Návrh prohlížeče formou zdi

5 Implementace aplikace

Tato kapitola se bude zabývat implementační částí aplikace, bude zmíněn princip sestrojení grafu scény, některé základní typy uzlů, manipulace s objekty, dále pak věci související přímo s během programu, jako dynamické vytváření a rušení objektů, animace apod. Některá fakta týkající se zejména teorie vychází ze zdroje [2].

Implementace této práce byla prováděna s využitím knihovny OpenSceneGraph (dále OSG) ve verzi 2.8.2 a jako vývojové prostředí sloužilo Microsoft Visual Studio 2008.

5.1 Sestrojení grafu scény

Jak bylo již zmíněno v kapitole 3, OSG uspořádává veškeré objekty do grafu scény skládajícího se z uzlů reprezentujících hlavní entity. Kořenový uzel se potom předává prohlížeči, který celým grafem kontinuálně prochází a zobrazuje jej. V prvé řadě bylo tedy potřeba vytvořit tento uzel. Jelikož obsahuje několik přímých potomků, měl by být skupinový. Veškeré další uzly se potom přidávají jako potomci tohoto kořene.

K umístění a vykreslení objektu na obrazovku je nutné použití a vzájemné provázání několika typů uzlů. Prvně je vhodné vytvořit instanci třídy `MatrixTransform`, která slouží pro manipulaci objektů v prostoru, a přiřadit jí jako rodiče kořenový uzel. Té se jako potomek přidává objekt `Geode`, který je listovým uzlem grafu a obsahuje informace o geometrii objektu. Data o geometrii jsou uložena ve třídě `Drawable`, která může představovat základní tvary, texty nebo přímo pole vrcholů objektu. Toto je tedy základní princip, jakým jsou objekty vykreslovány ve scéně.

Celé rozhraní se potom skládá jednak z jednotlivých položek na zdi, kterým je přiřazen obrázek a text a které jsou sjednoceny pod jedním objektem reprezentujícím tuto pomyslnou zeď, kvůli snadnému posouvání všech prvků záraz. Dále pak obsahuje karusel pro průchod jednotlivými kategoriemi, který je opět tvořen objekty s textem.

5.2 Transformace objektů

K manipulaci a umisťování objektů ve scéně lze v OSG využít několika tříd k tomu uzpůsobených. V této práci byla téměř vždy použita `MatrixTransform`, která provádí transformace za pomoci třídy `Matrix`, reprezentující matice. Matice o rozměrech 4x4 jsou v OSG základním datovým typem, podporujícím transformace jako posunutí, rotaci, měřítko nebo projekci. Jejich metody poskytují pohodlné rozhraní pro práci s nimi.

Při vytváření karuselu byla uplatněna třída `AutoTransform`, aby veškeré texty směřovali k pozorovateli. Zajišťuje totiž automatickou transformaci natočení objektu směrem ke kameře.

5.3 Běh aplikace

Při samotném běhu aplikace se kontinuálně prochází graf scény v nekonečné smyčce. Ta je klíčová nejen pro samotné zobrazování objektů, ale také pro další úkony prováděné dynamicky.

5.3.1 Průchody grafem scény

Průchod grafem probíhá následovně. Začíná u kořenového uzlu, poté se rekurzivně prochází jednotlivé uzly potomků směrem dolů, dokud se nenarazí na uzel bez potomků, v tom případě se vrací na poslední ne zcela prozkoumaný a děj se opakuje. Tento způsob se nazývá prohlédávání do hloubky (angl. *depth-first search*). OSG provádí několik typů průchodů. Prvním z nich je *průchod událostí* (angl. *event traversal*), který zpracovává vstup myši a klávesnice a jiné uživatelské události. Dalším je *průchod aktualizací* (angl. *update traversal*), ten umožňuje úpravu grafu, jako přidávání nových uzlů nebo změnu geometrie, a mimo jiné vykonávání *callbacků* (speciálních funkcí, o kterých bude zmínka v příští části). Při *vyřazování* (angl. *cull traversal*) se testuje, zda se bude daný uzel vykreslovat v rámci viditelného okna, nepotřebné uzly vyřadí ze seznamu výstupního optimalizovaného grafu pro interní renderování. Posledním průchodem je *vykreslování* (angl. *draw traversal*), ve kterém jsou vykreslovány uzly tohoto výstupního seznamu pomocí volání OpenGL API funkcí.

5.3.2 Animace

Aby byly transformace objektů plynulé a aby bylo patrné, odkud a kam se například nějaký objekt pohybuje, jejich průběh se animuje. Pokud ovšem chceme docílit postupného vykreslování pohybu, je nutné aktualizovat polohu objektu během aktualizace grafu scény (angl. *update traversal*). To je provedeno s použitím speciálních tříd, tzv. *update callbacků*, neboli funkcí zpětných volání, které se provádí právě v rámci průchodu grafem. Každému objektu se přiřazuje vlastní callback, a tak je možné s nimi zvlášť kontinuálně provádět jisté úkony. Operace obsažené v těle těchto funkcí se ovšem provádějí od jejich vytvoření do konce programu, nebo pokud nejsou z objektu odebrány. Aby tedy mohly posloužit pro animaci jen po určitou dobu, je nutné jejich vykonávání nějakým způsobem pozastavit.

K tomu bylo využito globálních proměnných, jelikož jinak je dodatečný přímý přístup do callbacků takřka nemožný. Ty pak slouží jako příznaky, které určují kdy a jakou část funkce právě provádět. Tyto proměnné v naprosté většině nastavovány v tzv. *event handleru*, speciální třídě obsluhující události uživatelské interakce. Jejich podstata bude popsána [dále](#). Vykonávání těla callbacku je tedy implicitně pozastaveno a pouze v případě, že je nastaven příznak, začne se provádět příslušná transformace. Po jejím dokončení většinou funkce příznak sama „deaktivuje“.

Samotná animace potom pracuje na prostém principu. Opakovaně se nastavují jednotlivé transformační matice a na základě vnitřního pomocného počítadla se kontroluje počet, resp. doba těchto transformací. Jakmile se dosáhne kýžené pozice, funkce je pozastavena a počítadlo vynulováno pro příští použití. Výjimkou jsou pouze animace, které probíhají oběma směry, tedy je vhodné zachovat si aktuální stav.

V implementovaném rozhraní jsou využity například pro detaily položek nebo pro posun samotné zdi. Také kamera využívá tohoto typu animace k dosažení přirozenějšího vjemu pro uživatele. Na obrázku 5.1 je vidět položka v detailu, jejíž posun na tuto pozici je plynule vykreslován.



Obrázek 5.1 – Detaily souborů

5.3.3 Obsluha událostí

Každé grafické uživatelské rozhraní je založeno na interakci s uživatelem, která se provádí pomocí tzv. událostí. Existuje mnoho událostí korespondujících s operacemi uživatele. Například pohyb kurzorem myši, kliknutí myši, stisk tlačítka na klávesnici nebo změna velikosti okna. Pro tyto účely poskytuje OSG třídu `GUIEventHandler`. Ta se přiřazuje prohlížeči scény a je, podobně jako callback, volána během každého průchodu grafem scény.

Uvnitř metody této třídy se pak dělí události podle svého typu. Ten může být mimo jiné *KEYDOWN* (stisknutí tlačítka), *PUSH* (kliknutí myši), *SCROLL* (otočení kolečkem myši) nebo *FRAME*, který označuje událost vyskytující se každé vykreslení obrazovky. V rámci těchto typů se pak již určuje, jaká konkrétní klávesa byla stisknuta a podobně, a jaká bude tedy provedena obsluha.

V našem konkrétním rozhraní se obsluhují následující uživatelské operace. Když se na klávesnici stiskne šipka doprava, resp. doleva, posune se zeď opačným směrem, aby bylo možné vidět další objekty. Stiskem mezerníku se pak zeď přesune do původní polohy. Otáčení karuselu se provádí kolečkem myši. Dále se pak při kliknutí levým nebo pravým tlačítkem myši vykreslí objekt na zdi v detailu, resp. se provede výběr další kategorie z karuselu. A v neposlední řadě se událostí typu *FRAME* obsluhuje situace, kdy je kurzor myši nad nějakým objektem, který se mírně posune blíže ke kameře, aby měl uživatel odezvu na možnost jeho výběru.

5.3.4 Dynamické vytváření a rušení objektů

Za běhu programu se zobrazují jednak nové karusely s obsahem aktuální složky, jednak také nové objekty souborů ve zdi. Z důvodů úspory paměti není vhodné, aby byly všechny objekty alokovány při startu programu a zůstaly pouze do určité doby skryty. Je nutné tedy vytvářet tyto prvky dynamicky. To s sebou ovšem nese jistá úskalí. OSG totiž využívá pro průchod grafem scény iterátory, které v něm při změně struktury grafu nedokážou pokračovat. Z toho důvodu tedy není možné jednoduše při potřebě přidání objektů provést volání funkce z jakéhokoli místa v programu. Je proto nutné provádět vytváření nových prvků mezi jednotlivými průchody grafem přímo v zobrazovací smyčce instance prohlížeče třídy `osgViewer`.

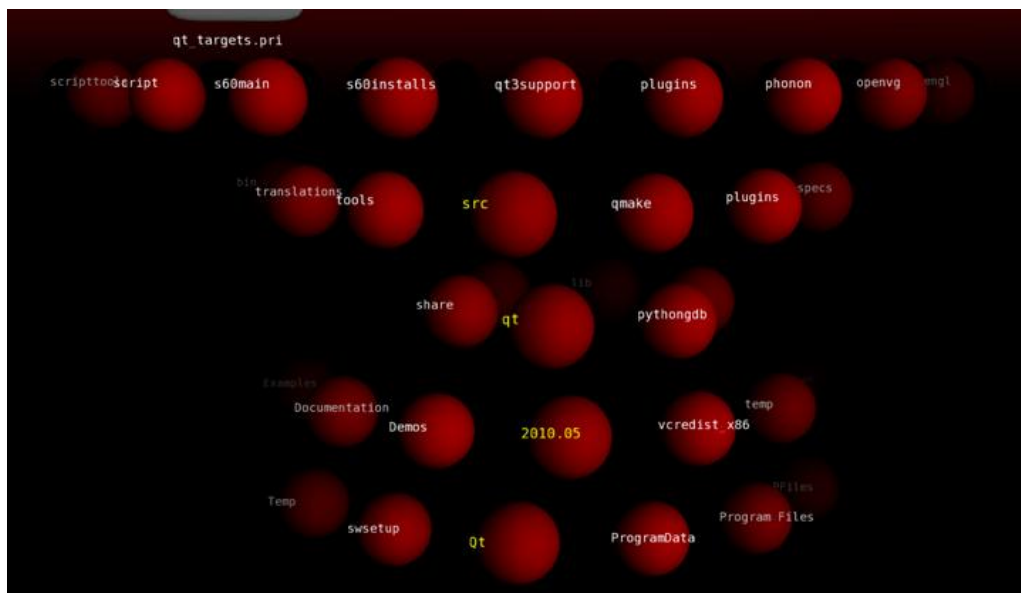
Podobně funguje i proces rušení objektů, který musí být také prováděn v určitou dobu. Samotné rušení bylo potom provedeno následovně. Nejprve se na základě ukazatele na objekt vyhledal jeho index v kořenovém uzlu (jeho rodič) a poté se zavolala funkce na odstranění potomka se zadaným indexem. Jakmile element není připojen ke zbytku grafu, není již vykreslován a paměť jím zabírána je dealokována.

5.3.5 Hierarchie souborů

Jak již bylo řečeno, k demonstraci hierarchické struktury souborů byla využita data přímo ze souborového systému počítače, na kterém aplikace běží. Jelikož byla vyvíjena pro operační systém Windows, byl jako kořenový uzel stromové struktury vybrán pevný disk C : . Ten je tedy postupně procházen a jeho obsah je prezentován s použitím implementovaného rozhraní.

Pro získání reálných dat bylo využito přímo funkcí od společnosti Microsoft [7], které jsou schopny po zadání cesty ke konkrétnímu adresáři zjistit veškerý jeho obsah, tedy seznam složek a souborů a k nim základní informace o jejich velikosti atd. Bylo ovšem potřeba vyřešit několik problémů, které se během vývoje vyskytly. Funkce například vyhledaly i složky, které zdánlivě neexistovaly, tedy nebyly zobrazitelné například v Průzkumníku, a při pokusu o přístup do nich spadl program s chybou. Aby se předešlo těmto pádům a aby tedy uživateli nebyly předkládány adresáře, které by potom nemohl dále procházet, je každá složka před zařazením do karuselu znovu předána funkci na vyhledání jejího obsahu a pokud skončí s chybou, je při vykreslování vynechána.

Bylo také potřeba udržovat informaci cestě k aktuálně procházenému adresáři. V případě, že se pak uživatel kupříkladu vrací ve struktuře o několik úrovní zpět, je cesta adekvátně upravena a je zobrazen příslušný obsah. Pro tvorbu samotných geometrických modelů souborů a složek bylo využito programu *3D Studio Max*¹.



Obrázek 5.2 – Karusely složek

¹ <http://usa.autodesk.com/3ds-max/>

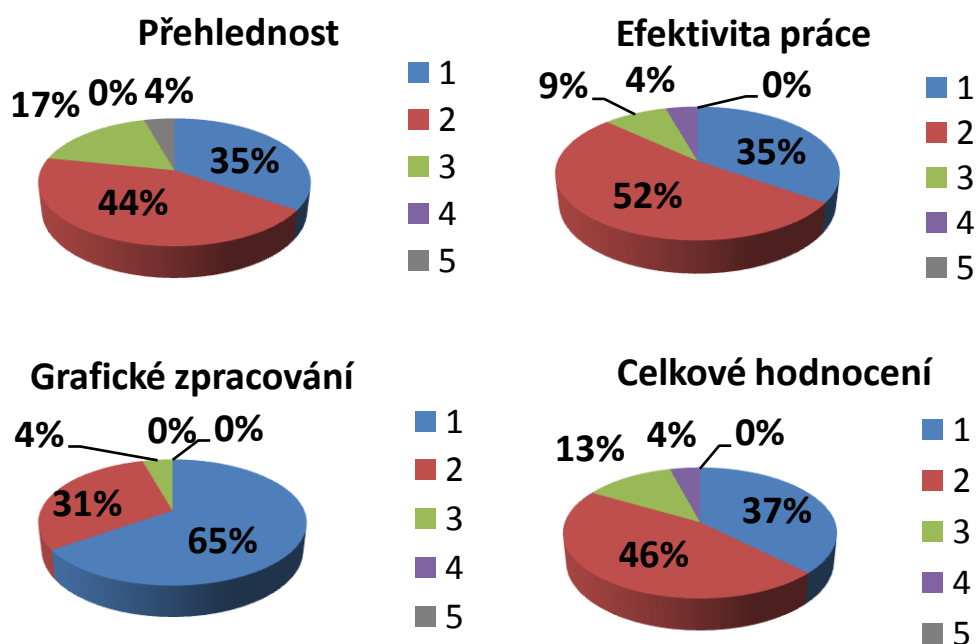
6 Testování

Výsledné rozhraní bylo podrobeno testování, které mělo ověřit jeho použitelnost a intuitivnost. Bylo osloveno 23 uživatelů nejčastěji kolem věku 22 let, kteří měli provádět předem definované úkony a na závěr vyjádřit svůj názor pomocí krátkého dotazníku, ve kterém hodnotili jednotlivé aspekty na stupnici od 1 do 5, přičemž 1 bylo nejvyšší hodnocení (viz Graf 6.1).

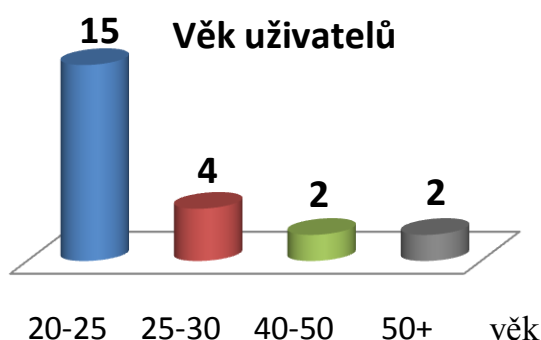
6.1 Uživatelské testy

V první řadě bylo testováno zorientování uživatele v rozhraní. Při jejich pozorování bylo zjištěno, že většina z nich měla zhruba první minutu problémy s polohou kamery, zeď se soubory se jim posouvala jinak, než zamýšleli a podobně. Po nějaké chvíli, když se s rozhraním více seznámili, jim již nečinilo vybrání konkrétního souboru či složky žádné potíže, také celkový princip procházení souborové struktury byl bezproblémový. Samozřejmě uživatelé ze starší věkové kategorie se v prostoru orientovali obtížněji a ovládnutím si nebyli příliš jisti.

Dále pak bylo uživatelům zadáno, aby vyhledali konkrétní soubor, ke kterému jim byla poskytnuta cesta. Jelikož jsou složky v karuselu seřazeny abecedně, tento úkol nedělal prakticky nikomu problémy a doba jeho provedení byla téměř srovnatelná s klasickým 2D přístupem, konkrétně v porovnání s Průzkumníkem ve Windows byl naměřený čas v průměru jen o několik málo sekund delší. Zejména při větším počtu položek v karuselu, kdy nejsou všechny viditelné, zabere jejich průchod více času. Naproti tomu díky větším rozměrům písma a všech objektů je rozhraní vhodné a pohodlné pro osoby s vadou zraku. Dalším aspektem, který urychlí orientaci, je jednotný styl zobrazení obsahu, zatímco v již zmiňovaném Průzkumníku si jednotlivé složky „pamatuji“ poslední zvolený styl, tudíž jsou uživatelé předkládány v mnoha podobách. Ani při úkolu, kdy se měly porovnat dva zadané obrázky v detailu, neměl nikdo větší potíže, jen bylo navrženo zlepšení v podobě možnosti zprůhlednění elementů v detailu pro snadnější výběr položek za ním.



Graf 6.1 a) b) c) d) – Hodnocení rozhraní uživateli (1 pro nejlepší)



Graf 6.2 – Věk testovaných uživatelů

6.2 Zhodnocení

Celkový dojem z rozhraní byl vcelku pozitivní, velká část mladších respondentů se již s rozhraním podobného stylu setkala, nicméně s podporou procházení hierarchických struktur nikoliv. Z dotazníku vyplynulo, že některým uživatelům přišlo poněkud nepřehledné, nicméně celkové grafické zpracování bylo ve velké většině hodnoceno velmi kladně. Na prostou otázku, jak by jej charakterizovali, odpovídali slovy: přehledné, jednoduché, „kukatelné“, použitelné, ale našli se i taci s přesnými opaky. Někteří uživatelé totiž neradi přechází k novým přístupům, nechtějí se přizpůsobovat. Ovšem ti, kteří rádi experimentují a zkoušejí nové metody tradičních činností, mohou takovýto přístup ocenit.



Obrázek 6.3 – Prohlížeč hierarchických struktur

7 Závěr

V této práci byly zmíněny některé přístupy k 3D uživatelským rozhraním. Byla popsána historie jejich vývoje a také možnosti jejich ovládání a zobrazování s pomocí moderních technologií pro interakci člověka s počítačem. Dále byla zmíněna návrhová, implementační a testovací fáze vývoje demonstrační aplikace, která ukazuje použití třetího prostoru na prohlížeči hierarchických stromových struktur. Snahou bylo dosáhnout netradičního, inovativního a efektivního přístupu. Jako případová studie byl zvolen konkrétně prohlížeč souborového systému, který byl reprezentován zdí souborů a otáčivým karuselem jednotlivých kategorií složek. Z výsledků testování vyplynulo, že většině uživatelů rozhraní vyhovuje a využili by ho v praxi.

Jako vstupní zařízení pro interakci s aplikací byla použita klasická myš, ovšem z hlediska charakteru ovládání by bylo možné použití také jiných metod, jako jsou dotykové displeje, datové rukavice, snímání pohybů uživatele a podobně. Co se týče dalšího vývoje, nabízí zvolený přístup nespočet možností. Bylo by například vhodné umožnit nejen procházení souborovým systémem, ale také přesuny a kopírování jednotlivých objektů, přímé přehrávání videí, úprava fotografií a podobně. V budoucnu by také mohlo být rozhraní zobrazováno na speciálních výstupních zařízeních, která by podpořila prostorový vjem manipulace s prvky. Své uplatnění by našlo kromě osobních počítačů také v mobilních telefonech, kde by bylo ovládáno kupříkladu pomocí polohového čidla, nebo ve webových aplikacích a portálech spravujících data v hierarchické struktuře. Pro tento účel by ovšem bylo vhodnější použití jiného vývojového nástroje než je OSG, jelikož je vcelku náročný na zdroje.

Literatura

- [1] *OpenSceneGraph website* [online]. 2011 [cit. 2011-04-14]. Dostupné z WWW: <<http://www.openscenegraph.org>>.
- [2] WANG, Rui; QIAN, Xuelei. *OpenSceneGraph 3.0 : Beginner's Guide* [online]. Birmingham : Packt Publishing Ltd., 2010 [cit. 2011-04-14]. Dostupné z WWW: <<http://www.packtpub.com/openscenegraph-3-0-beginners-guide/book>>. ISBN 978-1-849512-82-4.
- [3] BOWMAN, Doug A., et al. *3D User Interfaces : Theory and Practice*. Boston, Massachusetts : Addison-Wesley, 2005. 478 s. ISBN 0-201-75867-9.
- [4] MARTZ, Paul. *OpenSceneGraph Quick Start Guide : A Quick Introduction to the Cross-Platform Open Source Scene Graph API* [online]. Mountain View, California : Computer Graphics Systems Development Corporation, 2007 [cit. 2011-04-14]. Dostupné z WWW: <http://www.osgbooks.com/books/osg_qs.html>.
- [5] REIMER, Jeremy. *Ars Technica* [online]. 2005 [cit. 2011-04-14]. A History of the GUI. Dostupné z WWW: <<http://arstechnica.com/old/content/2005/05/gui.ars/>>.
- [6] *ViZoo : Advanced video design* [online]. 2011 [cit. 2011-04-14]. Dostupné z WWW: <<http://www.vizoo.com>>.
- [7] *MSDN Microsoft* [online]. Microsoft, 2011 [cit. 2011-04-14]. Listing the Files in a Directory. Dostupné z WWW: <<http://msdn.microsoft.com/en-us/library/aa365200.aspx>>.

Všechny odkazované online zdroje byly dostupné k 12.5.2011.

Seznam příloh

Příloha 1. Demonstrační plakát

Příloha 2. CD se zdrojovými texty, uživatelskou příručkou a dokumentací

Příloha 1

3D PROHLÍŽEČ HIERARCHICKÝCH STRUKTUR

Cílem práce bylo vytvořit 3D uživatelské rozhraní pro procházení hierarchických struktur. Bylo implementováno s využitím grafického souboru nástrojů

OpenSceneGraph. Rozhraní bylo demonstrováno na aplikaci, která umožňuje procházení souborového systému v operačním systému Windows. Jednotlivé soubory jsou zobrazovány v podobě zdi a složky pak jako položky otáčejícího se karuselu. Po výběru konkrétní složky se nový s aktuálním seznamem složek. Dále umožňuje zobrazení dvou elementů zdi v detailu vedle sebe pro možné úpravy fotografií a podobně.

Toto rozhraní je inovativním a perspektivním řešením procházení vcelku libovolných hierarchických struktur. Kromě klasického ovládání myší je možná jeho modifikace pro ovládání pohybem hlavy, datovou rukavici nebo přímo na dotykovém displeji například mobilního telefonu.



Bakalářská práce

Tomáš Hradečný

Vedoucí práce: Ing. Pavel Žák

Vysoké učení technické v Brně - Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
2011